February 03, 2016

MAPPING TOKENS TO IMAGES

Yuan Li

Benjamin Sapp

Attorney Docket No. LE-0551-01-US-DP

MAPPING TOKENS TO IMAGES

Ben Sapp

Yuan Li

ABSTRACT

[0001] The disclosure includes a system for mapping tokens to images to improve user queries for images. User queries, electronic documents with known relevance to each of the user queries, and a set of tokens are received by the system. The user queries and the electronic documents are mapped to the set of tokens by measuring an effectiveness of each token and electronic document combination based on a retrieval-performance metric. Search tokens for the electronic documents from the set of tokens are identified based on the mapping. The search tokens are output for each user query. The system advantageously retrieves documents for any token dictionary and directly optimizes image retrieval.

KEYWORDS

- image search
- image retrieval
- visual search
- visual token
- query expansion
- mapping tokens

BACKGROUND

[0002] Search engines may retrieve electronic documents by performing semantic entity searches. A search engine may use tokens to quickly retrieve search results based on user queries. Tokens include a limited, discrete dictionary of concepts that are indexed with each electronic document. For example, in a web search for webpages, tokens may correspond to words in the webpages. When a user provides a text-input search query, the search engine may map the search query to a set of tokens that are used to retrieve relevant electronic documents.

[0003] In visual search, the search engine may derive a set of tokens and tag images with the tokens after performing image processing on pixels in the images. For example, images of a dog, a cat, and a mouse may be identified based on pixel values indicative of different shapes, textures, colors, etc. The search engine may use entity-based visual retrieval tokens and natural language entity-resolution to map queries to visual tokens. Following with the example above, the images of the dog, the cat, and the mouse may be associated with the following visual tokens: dog_entity, cat_entity, and mouse_entity, respectively. If a user provides a natural language query of "photos of cute puppies," the search engine may parse the natural language input to infer that a relevant entity is dog_entity using entity resolution.

[0004] Semantic entity search may be limiting because the visual tokens can be more complex than semantic entity concepts. In addition, using natural language queries for visual search is problematic because such queries are optimized for text search and not visual search. As a result, in some cases, the visual search may retrieve unsatisfactory results.

DETAILED DESCRIPTION

[0005] A system of mapping tokens to images includes a search engine that determines search tokens that are used to tag an electronic document. The search engine receives user queries, electronic documents with known relevance to each of the user queries, and a set of tokens.

[0006] The user queries include text-input, such as natural language input. For example, the user queries may include variations of text from users searching for golden retrievers, such as "golden retriever," "golden retriever dog," etc. The electronic documents include images that match a user query and other documents that are used as a control. For example, the images may include matches for golden retrievers and other images selected randomly. Thus, the known relevance of each of the electronic documents may range from very relevant to not relevant.

[0007] A set of tokens may be derived from the pixel content for each image, for example, based on pixel analysis. The tokens may be part of any token dictionary. The tokens may be based on semantic entity concepts or may not map to a semantic entity concept.

[0008] The search engine maps the user queries and the electronic documents to the set of tokens by measuring an effectiveness of each token and electronic document combination based on a retrieval-performance metric. Each token combination may be considered effective if it maximizes the number of tokens that are associated with the images that match the user query and minimizes the number of tokens that are associated with the other images that do not match the user query. The retrieval-performance metric may include, for example, precision (the fraction of retrieved instances that are relevant) and recall (the fraction of all relevant instances that are retrieved). The retrieval-performance metric may also include f-measure, which combines precision and recall.

[0009] The search engine identifies search tokens for the electronic documents from the set of tokens based on the mapping. The search tokens may be identified as tokens in the set of tokens that have the highest performance based on the retrieval-performance metric. Continuing with the example above, the search tokens may include, for example, "golden retriever," "dog," "Labrador," and "animal." The search tokens may be associated with uniform weights or precision-ordered weights.

[0010] The search engine outputs the search tokens for each user query. The search engine may tag images with the search tokens and index them in a database. The search engine may subsequently receive a search query from a user and perform image retrieval by mapping the search query to the search tokens and retrieving the corresponding images from the database.

[0011] Figure 1 illustrates a diagram of an example visual search system 100 that includes a server 101, user devices 115a, 115n, and a network 105. The user devices 115a, 115n may access the server 101 via the network 105.

[0012] The server 101 may be a hardware device that includes a processor, a memory, and network communication capabilities. The server 101 may access the network 105 via signal line 102. The server 101 includes a search engine 109 and a database 199.

[0013] The search engine 109 can be code and routines for identifying search tokens that are relevant to images, tagging the images with the search tokens, indexing the images with the search tokens, and retrieving images from the database 199 based on user queries. In some implementations, the search engine 109 is implemented using hardware including a field-programmable gate array (FPGA) or an application-specific integrated circuit (ASIC). In

some implementations, the search engine 109 is implemented using a combination of hardware and software.

[0014] The user devices 115a, 115n may be computing devices that each include a memory and a processor, for example a laptop computer, a desktop computer, a tablet computer, a mobile telephone, a wearable device, a head-mounted display, a smart watch, a mobile email device, a portable game player, a portable music player, a reader device, a television with one or more processors embedded therein or coupled thereto, or other electronic device capable of accessing a network 105. The user devices 115a, 115n are communicatively coupled to the network 105 via signal lines 108 and 110, respectively. Users 125a, 125n interact with the user devices 125a, 115n, respectively.

[0015] A user 125 may input a search query into the user device 115 that is transmitted to the search engine 109. For example, the user 125 may access the search engine 109 via a browser 103 stored on the user device 115. In another scenario, the user device 115 may include a thin-client application that communicates with the search engine 109. The search engine 109 returns results to the user device 115 that are displayed via the browser 103 or as part of a thin-client application.

[0016] The network 105 can be a conventional type, wired or wireless, and may have numerous different configurations including a star configuration, token ring configuration or other configurations. The network 105 may include a local area network (LAN), a wide area network (WAN) (e.g., the Internet), and/or other interconnected data paths across which multiple devices may communicate. In some implementations, the network 105 may be a peer-to-peer network.

The network 105 may also be coupled to or includes portions of a telecommunications network for sending data in a variety of different communication protocols. In some implementations, the network 105 includes Bluetooth communication networks or a cellular communications network for sending and receiving data including via short messaging service (SMS), multimedia messaging service (MMS), hypertext transfer protocol (HTTP), direct data connection, WAP, email, etc.

[0017] Figure 2 is a flowchart of an example method 200 of mapping tokens to images that is discussed in conjunction with the example 300 illustrated in Figure 3. The method 200 of Figure 2 may be performed by the search engine 109 of Figure 1.

[0018] At block 202, user queries, electronic documents with known relevance to each of the queries, and a set of tokens are received. For example, Figure 3 illustrates electronic documents 305a, 305b, 305c, 305d, 305e, and 305f that correspond to [caesar salad]. Each of the electronic documents 305a, 305b, 305c, 305d, 305e, and 305f is associated with corresponding tokens 310a, 310b, 310c, 310d, 310e, and 310f. Tokens may be scored based on term frequency-inverse document frequency (tfidf) associated with the token. For example, the recall for a token may be multiplied by its log-precision to obtain the tfidf score. For each user query, a top number of tokens may be selected for analysis. For example, in the example illustrated in Figure 3, the top five tokens for [caesar salad] are {“salad”, “leaf vegetable”, “vegetable”, “dish”, “food”}. Using five tokens may sometimes be less effective, as is illustrated by this set due to overlap of coverage between tokens. For example, documents may have similar relevance to the terms “leaf

vegetable" and "vegetable." A larger set of tokens (e.g., the top 20 tokens) may offer a better balance between diversity of tokens and a manageable computational expense for the analysis.

[0019] At block 204, the user queries and the electronic documents are mapped to the set of tokens by measuring an effectiveness of each token and electronic document combination based on a retrieval-performance metric. For example, for each of the top 20 tokens, any combination of tokens that includes between one and five tokens may be examined for effectiveness. The precision and recall for each combination may be determined based on the electronic documents with known relevance and further, based on a set of randomly selected images (e.g., a set that includes about 100,000 images). In this example, 22,000 combinations were processed at about one second per combination. In instances where more combinations are analyzed, such as combinations of the top 30 tokens or when bigrams are used for increased precision, a greedy combination-finding approach may be used that analyzes a partial token combination, selects a next token that further improves the precision-recall point of the combination, and repeats the approach.

[0020] At block 206, search tokens are identified for the electronic documents from the set of tokens based on the mapping. For example, the combination with the maximum recall may be selected as search tokens. The combinations may be subject to a precision threshold, such as 0.5, 0.25, etc. Continuing with the example illustrated in Figure 3, the search tokens for [caesar salad] may be {"salad", "leaf vegetable", "poultry", "italian food"}. In some examples, the search tokens are associated with a uniform weight. In these examples, query results may

sometimes include too many results for images associated with certain tokens (e.g., "poultry") and not enough results for images associated with other tokens (e.g., "salad").

[0021] In some examples, search tokens are ordered based on precision for each token, and assigned decreasing weights. For example, for the user query [caesar salad] the search tokens may be {"salad": 0.5, "leaf vegetable": 0.25, "poultry": 0.125, "italian food": 0.0625}. As a result, a first image matching "salad" is ranked over a second image that does not match "salad," even if the second image matches some or all the other search tokens. If the search tokens are too precise and therefore return a limited number of results, a generic token, e.g. with low precision / high recall, may be added to the set. For example, "food" may be added to the search tokens and associated with a weight smaller than the weights for other search tokens.

[0022] At block 208, the search tokens for each user query are output. The system advantageously retrieves documents for any token dictionary and directly optimizes image retrieval. As a result, the search results may be more accurate than those produced by other retrieval systems.

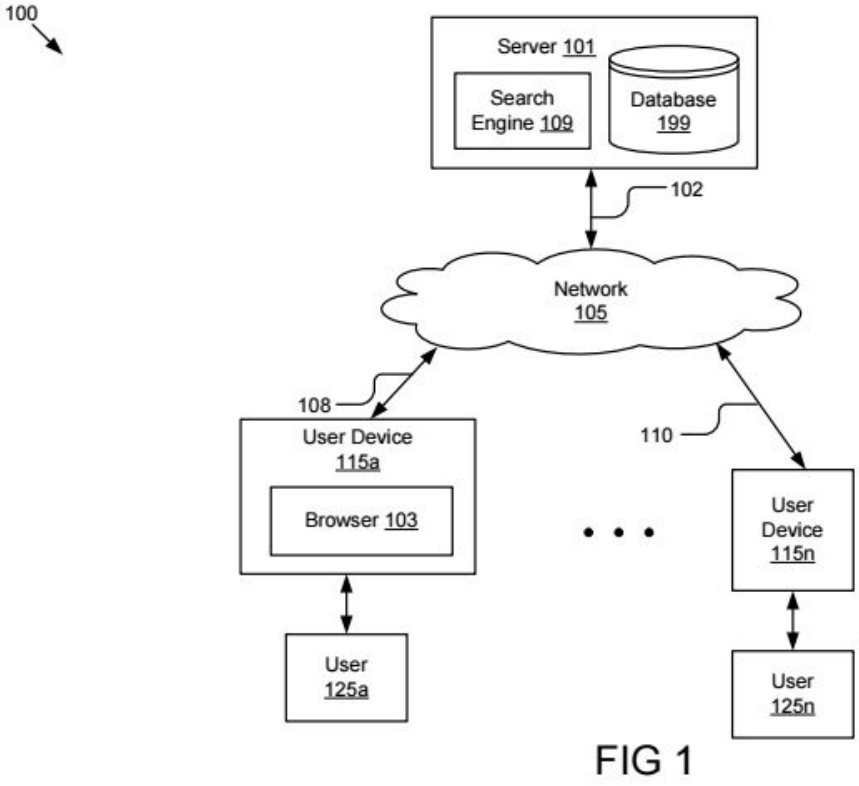
100
Server 101
Search Engine 109
Database 199
102
Network 105
108
User Device 115a
Browser 103
110
User Device 115n
User 125a
User 125n

FIG 1

LE-0551-01-US-NP
Sheet 2 of 3

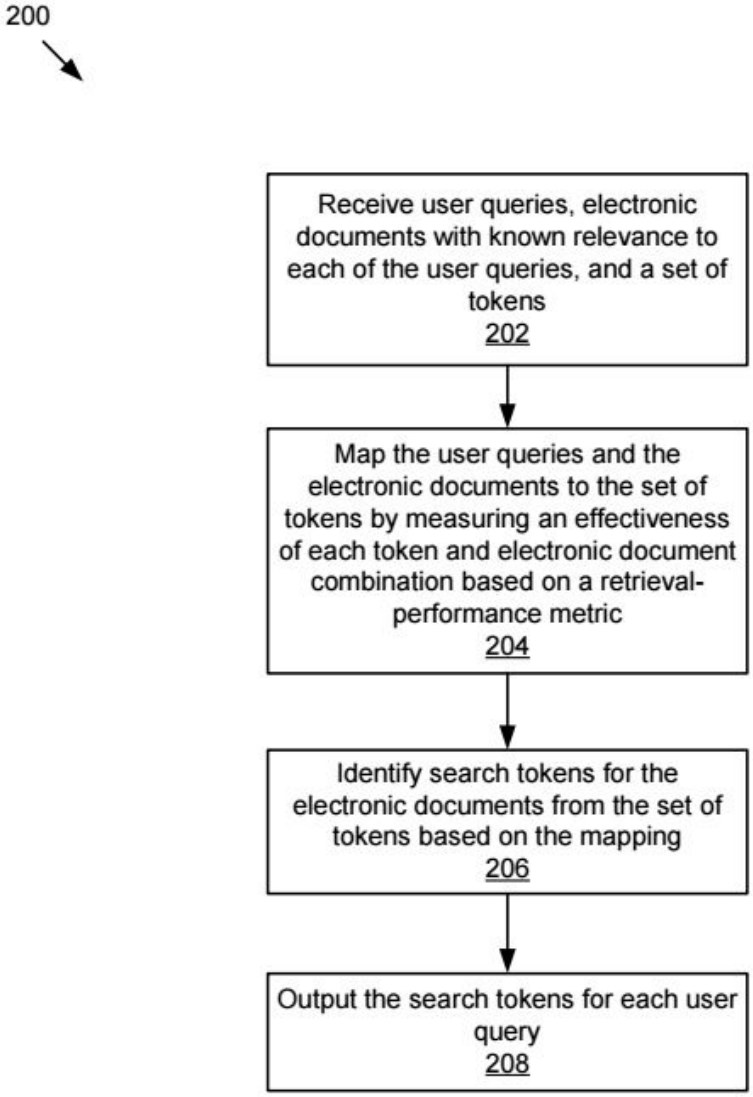

FIG 2

10

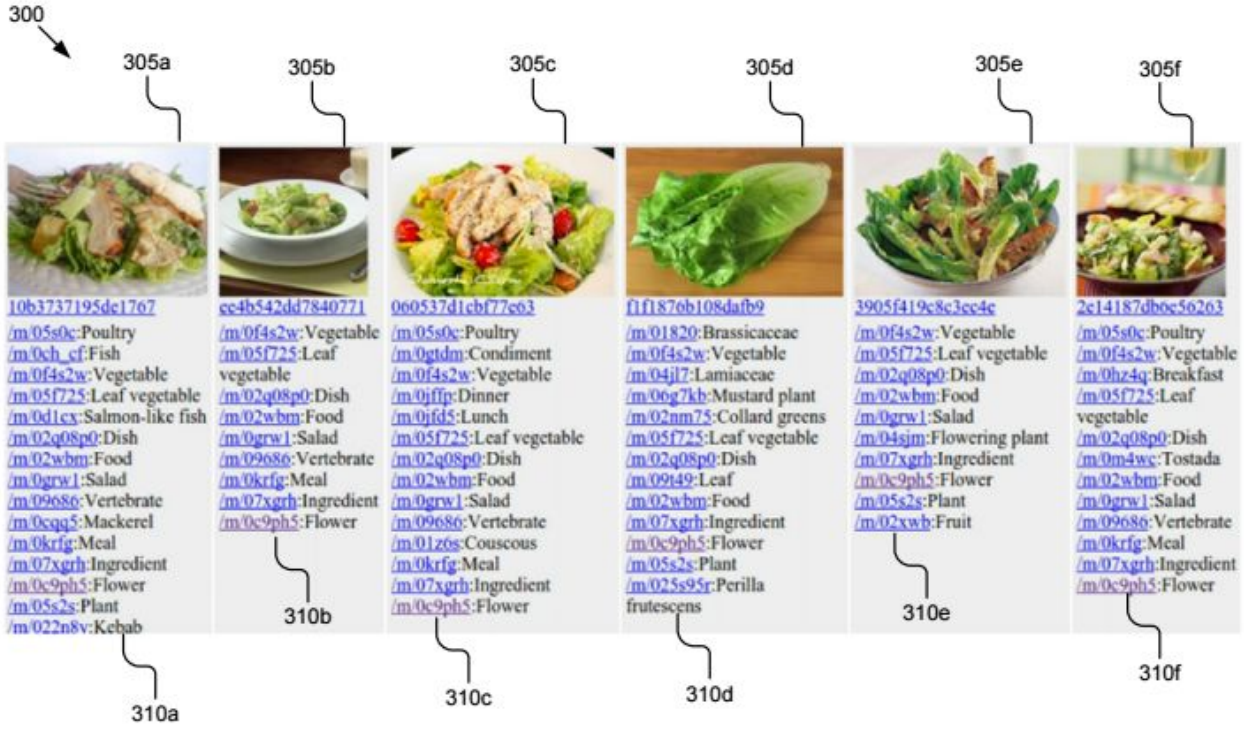
300
305a
305b
305c
305d
305e
305f
10b3737195de1767
/m/05s0c:Poultry
/m/0ch_cf:Fish
/m/0f4s2w:Vegetable
/m/05f725:Leaf vegetable
/m/0d1cx:Salmon-like fish
/m/02q08p0:Dish
/m/02wbm:Food
/m/0grw1:Salad
/m/09686:Vertebrate
/m/0cqq5:Mackerel
/m/0krfg:Meal
/m/07xgrh:Ingredient
/m/0c9ph5:Flower
/m/05s2s:Plant
/m/022n8v:Kebab
310a
ec4b542dd7840771
/m/0f4s2w:Vegetable
/m/05f725:Leaf vegetable
/m/02q08p0:Dish
/m/02wbm:Food
/m/0grw1:Salad
/m/09686:Vertebrate
/m/0krfg:Meal
/m/07xgrh:Ingredient
/m/0c9ph5:Flower
310b
060537d1cbf77e63
/m/05s0c:Poultry
/m/0gtdm:Condiment
/m/0f4s2w:Vegetable
/m/0jffp:Dinner
/m/0jfd5:Lunch
/m/05f725:Leaf vegetable
/m/02q08p0:Dish
/m/02wbm:Food
/m/0grw1:Salad
/m/09686:Vertebrate
/m/01z6s:Couscous
/m/0krfg:Meal
/m/07xgrh:Ingredient
/m/0c9ph5:Flower
310c
f1f1876b108dafb9
/m/01820:Brassicaceae
/m/0f4s2w:Vegetable
/m/04jl7:Lamiaceae
/m/06g7kb:Mustard plant
/m/02nm75:Collard greens
/m/05f725:Leaf vegetable
/m/02q08p0:Dish
/m/09t49:Leaf
/m/02wbm:Food
/m/07xgrh:Ingredient
/m/0c9ph5:Flower
/m/05s2s:Plant
/m/025s95r:Perilla frutescens
310d
3905f419c8c3cc4c
/m/0f4s2w:Vegetable
/m/05f725:Leaf vegetable
/m/02q08p0:Dish
/m/02wbm:Food
/m/0grw1:Salad
/m/04sjm:Flowering plant
/m/07xgrh:Ingredient
/m/0c9ph5:Flower
/m/05s2s:Plant
/m/02xwb:Fruit
310e
2c14187db6c56263
/m/05s0c:Poultry
/m/0f4s2w:Vegetable
/m/0hz4q:Breakfast
/m/05f725:Leaf vegetable
/m/02q08p0:Dish
/m/0m4wc:Tostada
/m/02wbm:Food
/m/0grw1:Salad
/m/09686:Vertebrate
/m/0krfg:Meal
/m/07xgrh:Ingredient
/m/0c9ph5:Flower
310f
LE-0551-01-US-NP
Sheet 3 of 3

FIG 3